

## **In the Claims**

1 1. (currently amended) A computer implemented method for solving a  
2 combinatorial optimization problem including a plurality of elements and a  
3 plurality of values, comprising:  
4       applying a priority algorithm in a form of an ordering function to an  
5 instance of the combinatorial optimization problem to produce an ordering  
6 of the elements;  
7       modifying the ordering of the elements to produce a re-ordering of the  
8 elements;  
9       applying a placement function to map values to the corresponding  
10 elements of the re-ordering; and  
11       repeating the modifying and the applying until all elements have been  
12 placed to obtain a solution of the combinatorial optimization problem.

1 2. (currently amended) The method of claim 1, in which the priority  
2 algorithm is fixed.

1 3. (currently amended) The method of claim 1, in which the priority  
2 algorithm is dynamic.

1 4. (original) The method of claim 1, in which the re-ordering is within a  
2 predetermined distance of the ordering.

1 5. (original) The method of claim 4, in which the distance is a Kendall-tau  
2 distance.

1 6. (currently amended) The method of claim 1, in which the re-ordering uses  
2 a decision vector, and in which the ~~distance~~ decision vector has one field for  
3 each element of the order, each field determining a new order of the element  
4 in the re-ordering.

1 7. (original) The method of claim 1, in which the re-ordering is probabilistic.

1 8. (new) A computer program product storing a computer program which  
2 when executed by a computer performs a method for solving a combinatorial  
3 optimization problem including a plurality of elements and a plurality of  
4 values by performing the steps of:

5       applying a priority algorithm in a form of an ordering function to an  
6 instance of the combinatorial optimization problem to produce an ordering  
7 of the elements;

8       modifying the ordering of the elements to produce a re-ordering of the  
9 elements;

10       applying a placement function to map values to the corresponding  
11 elements of the re-ordering; and

12       repeating the modifying and the applying until all elements have been  
13 placed to obtain a solution of the combinatorial optimization problem.

1 9. (new) A computer implemented method for solving a combinatorial  
2 optimization problem including a plurality of elements and a plurality of  
3 values, comprising:  
4       applying a priority algorithm in a form of an ordering function to an  
5 instance of the combinatorial optimization problem to produce an ordering  
6 of the elements, in which the priority algorithm is dynamic;  
7       modifying the ordering of the elements to produce a re-ordering of the  
8 elements;  
9       applying a placement function to map values to the corresponding  
10 elements of the re-ordering; and  
11       repeating the modifying and the applying until all elements have been  
12 placed to obtain a solution of the combinatorial optimization problem.

1 10. (new) The method of claim 9, in which the re-ordering is within a  
2 predetermined distance of the ordering.

1 11. (new) The method of claim 10, in which the distance is a Kendall-tau  
2 distance.

1 12. (new) The method of claim 9, in which the re-ordering uses a decision  
2 vector, and in which the decision vector has one field for each element of the  
3 order, each field determining a new order of the element in the re-ordering.

1 13. (new) The method of claim 9, in which the re-ordering is probabilistic.